International Journal of Innovative Computing, Information and Control
Volume **14**, Number **3**, June **2018**
ICIC International ©2018 ISSN 1349-4198
pp. 997–1013

A SELF-ADAPTIVE SPATIAL-TEMPORAL CORRELATION PREDICTION ALGORITHM TO REDUCE DATA TRANSMISSION IN WIRELESS SENSOR NETWORKS

Junyong Ruan[1] and Zhao Lu[2]

[1]Department of Equipment
Qingdao First Sanatorium of the People's Liberation Army Navy
No. 27, Hongkong West Road, Qingdao 266000, P. R. China
ruanjunyongqdyl@163.com

[2]Beijing Kingsoft Cloud Network Technology Corporation Limited
No. 33, Xiaoying West Road, Qinghe Street, Beijing 100085, P. R. China

Received July 2017; revised December 2017

Abstract. *In this work, we propose a self-adaptive spatial-temporal correlation (SAS-TC) prediction algorithm, which combines temporal correlation and spatial correlation to prediction sensing data in wireless sensor networks. Adaptive grey prediction model is used to measure the temporal correlation of sensing data, and then Delaunay triangulation is introduced to measure the spatial correlation of sensing data. The adaptive grey prediction model runs between member nodes and cluster heads, using the temporal correlation of data within these nodes and heads to reduce the amount of data transmission. The spatial correlation model runs between cluster heads and sink nodes, using the spatial correlation of data within the member nodes to reduce the amount of data transmission. The simulation results show that the algorithm has higher prediction accuracy, which can effectively reduce the amount of data transmission in the network and save the energy consumption of data transmission.*
Keywords: Wireless sensor networks, Network lifetime, Data prediction, Self-adaptive spatial-temporal correlation

1. **Introduction.** Wireless sensor networks (WSNs) are deployed in a geographical area for monitoring physical phenomena like temperature, humidity, seismic events [1]. In order to obtain exact information of environments or events, a large number of sensing nodes are deployed to collect and report data to sink nodes at a high frequency rate. The data generated by sensor nodes usually have high spatial-temporal correlation and contain large amounts of redundant data. Meanwhile, transferring redundant data will cause unnecessary energy consumptions and collisions. Thus, it is a significant problem to reduce energy consumption and extend the lifetime of WSNs. Data prediction is an efficient technique for solving this problem. By exploring the spatial-temporal correlation, two synchronized predictors are used on both sensor nodes and sink nodes. If data prediction error is smaller than the given threshold, sensor nodes will not send the data to sink nodes. Sink nodes regard prediction values as sensing data, which can reduce the data to be transmitted and communication energy cost, and thus prolong the network's lifetime.

Temporal correlation prediction models in WSNs have been studied to prolong the lifetime of sensor networks, which mainly adopt typical time series models. Lazaridis and Mehrotra proposed a piecewise constant approximation (PCA) method which uses a constant value as the prediction value [2]. Mollanoori et al. proposed an online prediction framework and some other simple linear predictions, that is, single point predictor (SPP)

[3], simple linear extrapolation (SLE) predictor and even linear extrapolation (ELE) predictor. Lim and Shin [4] proposed two different kinds of self-adaptive linear predictors: directly smoothed slope based linear (DSSL) predictor and directly averaged slope based linear (DASL) predictor. Gaura et al. [5] proposed a linear Spanish inquisition protocol (L-SIP), which uses double exponentially weighted moving average (EWMA) to make the prediction. Han et al. [6] proposed that each node associates with an upper bound and a lower bound, and these bounds instead of the raw data are sent to sink nodes. Similarly, McConnell and Skillicorn [7] proposed that each sensor transmits the predicted target class rather than the raw data to sink nodes. These two approaches are classified into algorithmic techniques, but their computational and transmission complexity is high. Singh et al. used grey prediction algorithm to predict data based on the sliding of historical data segments [8]. Moving average prediction algorithm, autoregressive prediction algorithm and autoregressive moving average prediction algorithm belong to typical prediction algorithms based on time correlation [9]. They can be used in many practical cases with good accuracy, but as we all know this is an impossible task for nodes that have limited computing capacity and memory to run a complex long-term prediction algorithm. In addition, these algorithms based on time correlation require nodes to store a large amount of historical data, which requires a large storage space for nodes. Moreover, these algorithms have low prediction accuracy when the data fluctuate, so they are only suitable for the data with higher time correlation. Therefore, in order to avoid such limitations, it is necessary to develop an efficient data prediction method in WSNs [10].

In fact, WSNs have the characteristic of a large scale intensive deployment of nodes, which often results in a strong spatial correlation between the monitoring data of adjacent nodes [11,12]. Spatial correlation refers to the fact that certain physical phenomena observed by the nodes are spatially consistent, resulting in the similarity of data collected by adjacent nodes [13]. Jiang et al. [14] proposed an adaptively enabling/disabling prediction scheme for clustering which is based on an AR prediction model. Algorithmic approaches mainly exploit the heuristic or behavioral characteristics of the sensing phenomena in order to aggregate data. Typically, Goel and Imielinski [15] proposed a paradigm called the prediction-based monitoring for energy-efficient monitoring (PREMON) based on the concept of MPEG encoding and then proposed a buddy protocol [16], which extends the PREMON by using a distributed scheme to exploit spatial correlations between sensor nodes. The main drawback of these approaches is their considerable high computational cost.

Therefore, combining temporal correlation with spatial correlation to make data prediction can solve the shortcoming of temporal correlation prediction methods especially when the data fluctuate. Choi et al. [17] presented a cost effective monitoring scheme for cyber-physical system platform using a spatio-temporal model. Chen et al. [18] developed a clustered spatio-temporal compression scheme by integrating network coding (NC), compressed sensing (CS) and spatio-temporal compression for correlated data. Kandukuri et al. [19] proposed a hybrid data aggregative window function (DAWF) algorithm for exploiting both spatial and temporal data redundancies in WSNs [20]. The spatial-temporal correlation is an important feature of sensor data in WSNs. Based on the analysis of spatial-temporal correlation in WSNs, in this work we propose a self-adaptive spatial-temporal correlation (SASTC) prediction algorithm, which introduces the Delaunay triangulation diagram to calculate the spatial correlation weighting factors for monitoring data in the network, and uses the Markov process to accurately describe the changing process of monitoring data from adjacent nodes in WSNs. This algorithm is suitable for clustered network and mainly consists of two parts: adaptive grey prediction model and spatial correlation prediction model. The adaptive grey prediction model runs between member nodes and cluster heads, using the temporal correlation of data within these nodes and heads to reduce the amount of data transmission. The spatial correlation model runs between cluster heads and sink nodes, using the spatial correlation of data within the member nodes to reduce the amount of data transmission. Simulation results show that the algorithm can effectively reduce a significant number of data transmissions.

The rest of the work is organized as follows. In Section 2, after briefly reviewing the focusing problem, we detail the self-adaptive temporal correlation (SATC) prediction model. In Section 3, the spatial correlation prediction model is introduced. In Section 4, we use simulation results to verify the proposed prediction model. Section 5 concludes the work.

2. **Self-Adaptive Temporal Correlation (SATC) Prediction Model.**

2.1. **Network model.** The research of this paper is based on clustering algorithms, such as low energy adaptive clustering hierarchy (LEACH), geographical adaptive fidelity (GAF), and TopDisc. After the completion of the clustering, cluster heads will build the Delaunay triangulation diagram according to node locations in the cluster. Delaunay triangulation has the feature of empty round and its least angle is the largest, which ensures adjacent graphs using the pattern are more uniform and reasonable compared to other forms of adjacent graphs [21,22]. Thus, using Delaunay triangulation to measure the spatial correlation is more accurate. As shown in Figure 1, the solid dots represent the cluster heads, the hollow dots represent member nodes in clusters, the dotted lines represent the scope of cluster, and the solid lines indicate the adjacent diagram of the Delaunay triangle network in the cluster.

Delaunay triangulation adjacent graph is the dual graph of Voronoi diagram. Nodes divide the whole monitoring area according to the neighboring principle in the network.

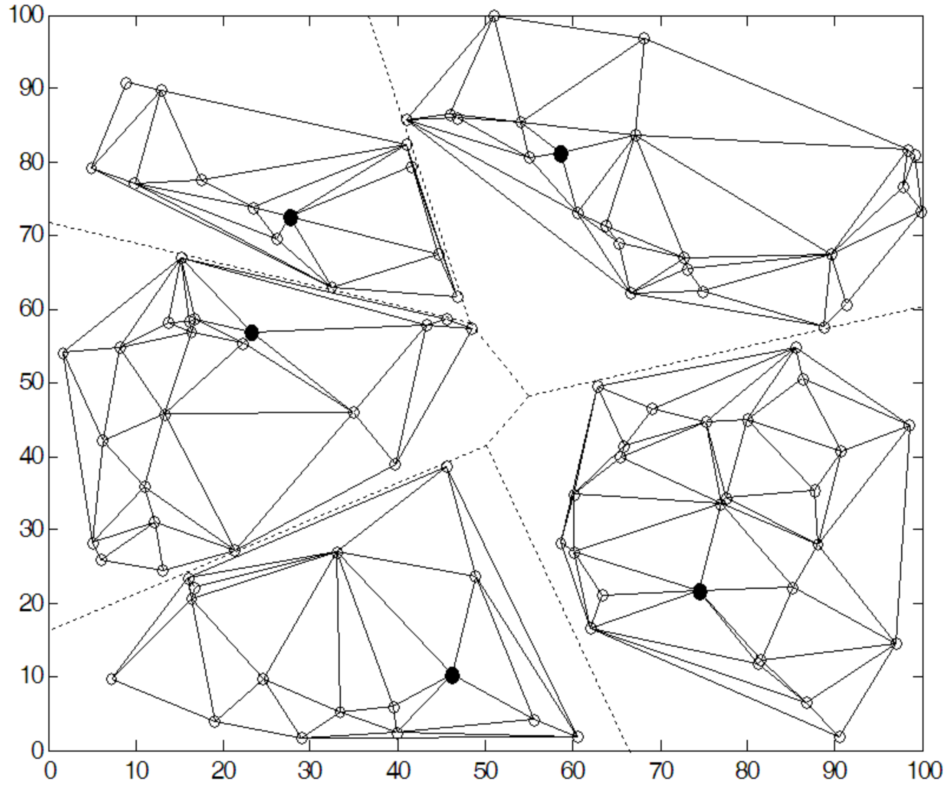

FIGURE 1. Delaunay triangulation graph of nodes in cluster

The continuous polygon composed by perpendicular bisector of the line between two adjacent monitoring regions is the Voronoi diagram [23]. Setting N as the set of finite nodes in area R^2, the Euclid distance of any position $s(s_1, s_2)$ in the monitoring region to node $n(n_1, n_2)$ is as follows:

$$\|n - s\| = \sqrt{(n_1 - s_1)^2 + (n_2 - s_2)^2}, \quad n \in N \tag{1}$$

For each node, the distance of any position within the monitoring area to the node is less than the distance from the position to other nodes:

$$\|n - s\| \le \|t - s\|, \quad \forall t \in N \tag{2}$$

The divided node monitoring area that is generated by the above method is the Voronoi diagram. Connecting two adjacent points within Voronoi diagram which have common boundary is Delaunay triangulation adjacent diagram [24]. This article uses Delaunay triangulation to construct adjacent graphs in the cluster network. It measures the spatial correlation coefficient according to the distance between adjacent nodes. Spatial correlation coefficient is used to determine the influence of changes between node data, which constitute one-step transition probability matrix of Markov chain prediction.

2.2. **Temporal correlation prediction model.** This section introduces the SATC prediction model that works in nodes and cluster heads. The main purpose of this model is to reduce the amount of data transmission from member nodes to cluster heads. Nodes analyze their short-term historical data to determine short-term trends, and send predicted values to the cluster heads. According to the predicted values, if prediction error is less than the given threshold which depends on application situations and experiences, node will not send data to cluster heads. This process can reduce node energy consumption. Moreover, if several prediction errors are greater than the given threshold, we can say the changing tendency of the data has changed. Then, nodes will adjust prediction parameters according to the short-term historical data, and send monitored values and adjusted parameters to clusters.

Setting S as the sensor network, it consists of a collection of n sensing nodes $S = \{s_1, s_2, \ldots, s_n\}$ and a sink node. All data generated by the sensor network S can be written as $V = \{V_1, V_2, \ldots, V_n\}$. $V_i = \{v_i(t_1), v_i(t_2), v_i(t_3), \ldots\}$ where $1 \le i \le n$ is a time sequence set generated by sensor node s_i in each T second. The whole sensor network is grouped into clusters. The nodes in the network send data to a gathering node using single hop or multi-hop. $P = \{P_1, P_2, \ldots, P_n\}$ is the prediction data generated by n sensor nodes in the sensor network S. $P_i = \{p_i(t_1), p_i(t_2), p_i(t_3), \ldots\}$ is the prediction data generated by sensor node s_i in each T second.

As shown in Figure 2, continuous 10000 temperature data were collected by a node over about five days, and the time interval of data collection is about 40 seconds. Through the picture we can see that the data changing trends during a specific period are roughly linear, but there are fluctuations above it. Thus, we cannot achieve ideal accuracy prediction if we only use a single linear prediction algorithm or nonlinear data algorithm. Meanwhile, some complex data prediction algorithms such as neural networks cannot be conducted in sensor nodes because of their limited computing power. Based on the analysis of data characteristics, this paper decomposes the changes of monitoring data into the linear component and the nonlinear component:

$$V_i = M_i + X_i \tag{3}$$

$$v_i(t) = m_i(t) + x_i(t) \tag{4}$$

where M_i is a linear trend component of node s_i that grows over time, $M_i = \{m_i(t - n + 1), m_i(t - n + 2), \ldots, m_i(t)\}$, and X_i is the nonlinear trend component of node s_i,

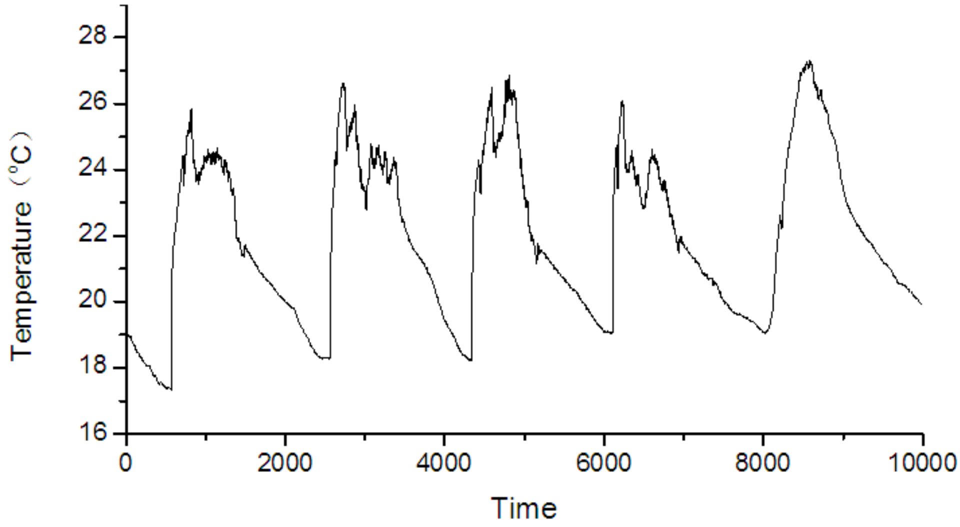

FIGURE 2. Data of temperature

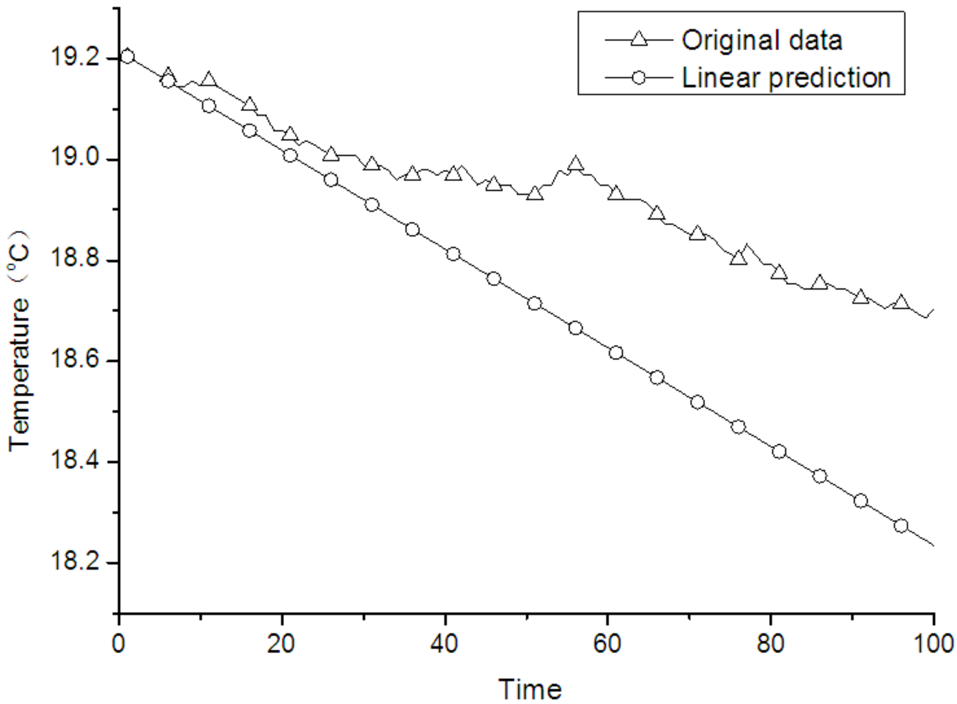

FIGURE 3. Original data and linear prediction

$X_i = \{x_i(t), x_i(t-1), x_i(t-2), \ldots\}$. The linear trend component is represented by a first order linear equation:

$$m_i(t) = \alpha_i + \beta_i t \tag{5}$$

where α_i and β_i are real variables which are determined by the least square method.

In parameter adjustment phase, nodes use the least mean square fitting method to calculate the parameters based on the change trends of historical data, and send predicted values to cluster heads. In most of computational software such as MATLAB, the corresponding function to the least square method is built, so it is easy to determine the values of α_i and β_i. Cluster heads and member nodes predict linear change component according to α_i and β_i. As shown in Figure 3, we can use the first four data to calculate

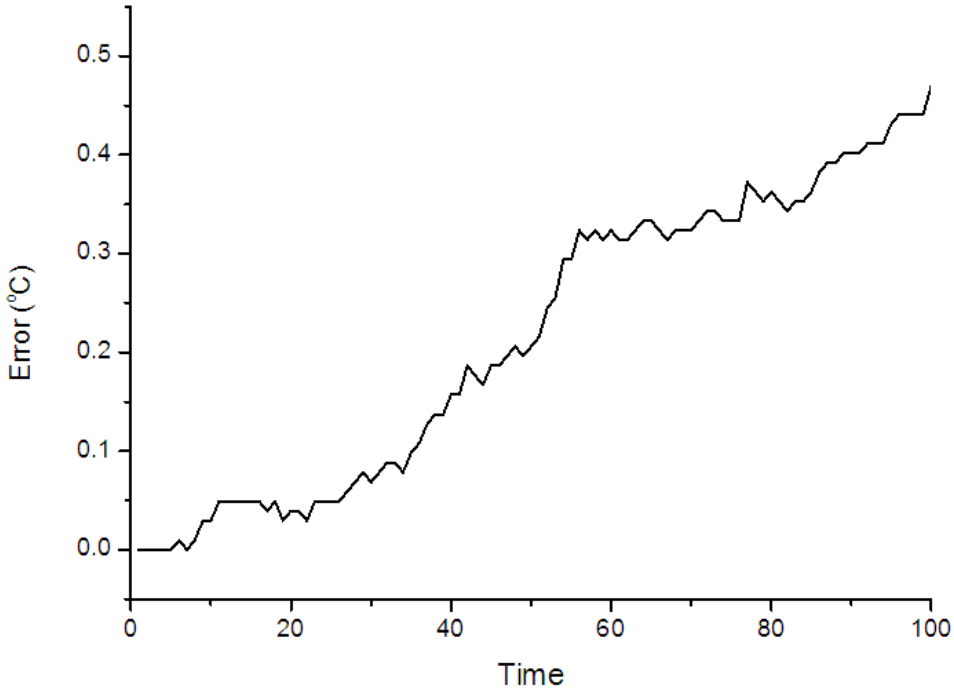

FIGURE 4. Prediction error change by the AR first order autoregressive model

the prediction parameters, and then predict the subsequent linear change part of the time series data according to α_i, β_i and Equation (5). Equation (3) and Equation (4) can be used to calculate the non-linear part of the subsequent data, as shown in Figure 4. In [25], authors use AR first order autoregressive model to predict the partial data changes. AR (autoregressive) model is suitable for the prediction of the linear stationary process.

$$x_i(t) = a_i x_i(t-1) + b_i x_i(t-2) + c_i x_i(t-3) + \delta_i N(0,1) \quad (6)$$

where a_i, b_i and c_i are real variables, and $\delta_i N(0,1)$ is the standard error function of white noise. However, with the changing of the data, the prediction error will increase constantly when using α_i and β_i to indicate the linear change, which leads to a growing number of nonlinear parts of the data. Thus, using the AR forecast model to predict the nonlinear changes is not suitable for data that have nonlinear increase or decrease characteristics. Otherwise, the prediction error is large, and we need to adjust the forecast parameters α_i, β_i, a_i, b_i and c_i frequently according to the data changing.

Grey prediction is suitable for predicting the nonlinear change of data. Grey mathematics is used to deal with uncertain and quantitative data. Grey mathematics regards the observed data sequence as grey or grey process that changes over time, and takes full advantage of the known data information to seek out the patterns of data changes. It generates corresponding data change model, and makes predictions through establishing data accumulation or regression. Grey prediction models can make accurate prediction by using a small number of historical data. Grey prediction is a kind of ideal prediction methods for a small amount of observation data with nonlinear growth. This paper uses the GM$(1,1)$ model which is a common grey prediction model to predict the nonlinear change part of data. GM$(1,1)$ model is a first order model which only contains one variable. The model is simple, but it has a good prediction effect for nonlinear data. Thus, the model is suitable for running in the nodes with limited computing power.

Set X_i as the original data series of GM$(1,1)$:

$$X_i = \{x_i(1), x_i(2), \ldots, x_i(n)\} \quad (7)$$

Set $X_i^{(1)}$ as the first order accumulation generating operator (AGO) series of X_i:

$$X_i^{(1)} = \left\{x_i^{(1)}(1), x_i^{(1)}(2), \ldots, x_i^{(1)}(n)\right\} \tag{8}$$

where

$$x_i^{(1)}(k) = \sum_{t=1}^{k} x_i(t), \quad k = 1, 2, \ldots, n \tag{9}$$

Set $Z_i^{(1)}$ as the generated mean sequence of consecutive neighbors of $X_i^{(1)}$:

$$Z_i^{(1)} = \left\{z_i^{(1)}(2), z_i^{(1)}(3), \ldots, z_i^{(1)}(n)\right\} \tag{10}$$

$$Z_i^{(1)}(k) = 0.5x_i^{(1)}(k) + 0.5x_i^{(1)}(k-1) \tag{11}$$

The grey differential equation model of GM$(1,1)$ is:

$$x_i(k) + az_i^{(1)}(k) = b \tag{12}$$

where a is the developing coefficient and b is the grey model parameter. Set u as the parameter vector to be estimated, then $u = (a, b)^T$. The least squares estimate parameter of Equation (10) satisfies the following equation:

$$\hat{u} = \left(B^T B\right)^{-1} B^T Y_n \tag{13}$$

$$b = \begin{bmatrix} -z^{(1)}(2) & 1 \\ -z^{(1)}(3) & 1 \\ \cdots & \cdots \\ -z^{(1)}(n) & 1 \end{bmatrix} \tag{14}$$

$$y = \begin{bmatrix} x^{(0)}(2) \\ x^{(0)}(3) \\ \cdots \\ x^{(0)}(n) \end{bmatrix} \tag{15}$$

Albinism differential equation of GM$(1,1)$ can be expressed as:

$$\frac{dx_i^{(1)}}{dt} + ax_i^{(1)} = b \tag{16}$$

The solution of Equation (16) is:

$$\hat{x}_i^{(1)}(t) = \left(x_i^{(1)}(1) - \frac{b}{a}\right) e^{-at} + \frac{b}{a} \tag{17}$$

Time series of GM$(1,1)$ is:

$$\hat{x}_i^{(1)}(k+1) = \left[x_i^{(1)}(1) - \frac{b}{a}\right] e^{-ak} + \frac{b}{a}, \quad k = 1, 2, \ldots, n \tag{18}$$

and $x_i^{(1)}(1) = x_i(1)$, then

$$\hat{x}_i^{(1)}(k+1) = \left[x_i(1) - \frac{b}{a}\right] e^{-ak} + \frac{b}{a}, \quad k = 1, 2, \ldots, n \tag{19}$$

The prediction can be computed as:

$$\begin{aligned}\hat{x}_i(k+1) &= \hat{x}_i^{(1)}(k+1) - \hat{x}_i^{(1)}(k) \\ &= \left[x_i(1) - \frac{b}{a}\right] e^{-ak} + \frac{b}{a} - \left[x_i(1) - \frac{b}{a}\right] e^{-a(k-1)} + \frac{b}{a} \\ &= \left[x_i(1) - \frac{b}{a}\right] \left(e^{-ak} - e^{-a(k-1)}\right)\end{aligned} \tag{20}$$

Equation (20) is the prediction equation of the nonlinear part. The prediction value of node s_i at time t is:

$$p_i(t) = \hat{m}_i(t) + \hat{x}_i(t) \tag{21}$$

The prediction error of node s_i at time t is:

$$e_i(t) = v_i(t) - p_i(t) \tag{22}$$

Based on the above analysis, the basic process of applying GM$(1,1)$ into the prediction of nonlinear changes in the sensing data is as follows. Nodes create a sequence of length n during every T interval, $V_i = \{v_i(t-n+1), v_i(t-n+2), \ldots, v_i(t)\}$. Every node uses minimum mean square to fit the parameters α_i and β_i according to Equation (5) in the parameter adjustment phase, and calculates subsequent linear series according to the parameters α_i and β_i. Then nodes make out sequence X_i according to Equation (4). At last, nodes use grey prediction model to calculate the coefficients a_i and b_i. Data prediction model running in nodes can uniquely determine predictive parameters α_i, β_i, a_i and b_i. Then, nodes send four parameters to the cluster heads. Cluster heads can compute the value of these nodes according to α_i, β_i, a_i, b_i and their historical data.

2.3. **Simulation analysis of the temporal correlation model.** In order to validate the prediction effect of the temporal correlation model in 2.2, the data in Figure 2 are used. In this paper, the value of W is 2 in the simulation. The parameters are adjusted more frequently when data have larger fluctuations, as compared in Figure 2 and Figure 5. Compared with the ADC model which is proposed in [26], the SATC model not only needs

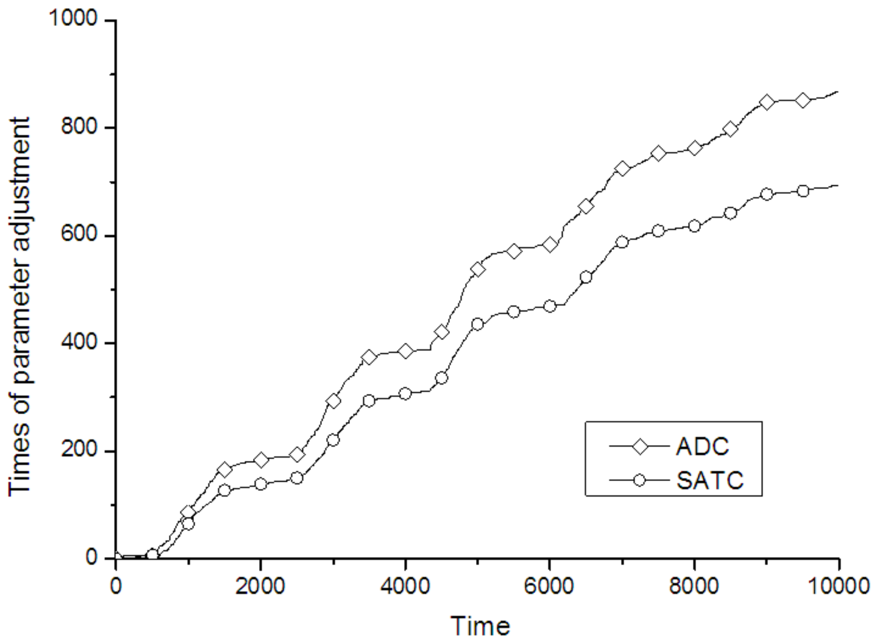

FIGURE 5. The time comparison of parameter adjustment using ADC and SATC

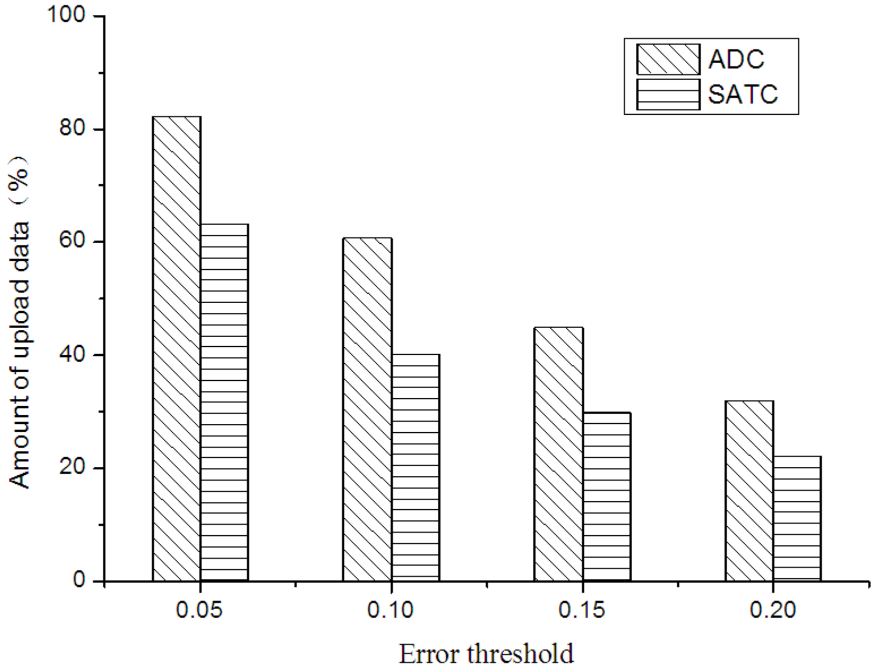

FIGURE 6. The effect comparison of ADC and SATC

less prediction parameters but also needs less time for parameters adjustment. Figure 6 shows the amount of data uploaded in the prediction process by the two algorithms. The amount of uploaded data in SATC model is lower than that in ADC prediction algorithm under different error thresholds. There are two reasons for this result, that is, fewer prediction parameters and less time for parameters adjustment.

3. The Spatial Correlation Prediction Model.

3.1. **Spatial correlation model.** This section mainly introduces the data prediction model between cluster heads and sink nodes. The main purpose of the model is to reduce the data transmission from member nodes to sink nodes. In this paper, we take the Delaunay triangulation of cluster nodes as $G(V, E)$, where $V = \{1, 2, 3, \dots, n\}$ represents the collection of nodes in the monitoring area, E represents a collection of neighboring nodes, and n represents the number of nodes in the network. The collection of neighbor nodes of node i is $N_i = \{j \in V | (i, j) \in E\} \subset V$, and the adjacency matrix of graph $G(V, E)$ is $A = [a_{ij}] \in R^{n \times n}$ where $a_{ij} = 1$ when $(i, j) \in E$ and otherwise $a_{ij} = 0$.

The number of elements in the collection N_i is called the degree of node i, denoted as l_i, and then $l_i = \sum_{j=1}^{n} a_{ij}$. According to the property of the Delaunay triangulation, the node i has at least two adjacent nodes, so $l_i \geq 2$.

Let d_{ij} represent the Euclidean distance between node $i(x_i, y_i)$ and node $j(x_j, y_j)$, and then $d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$. dm_i is the distance between node i and its nearest neighbor nodes, namely, $dm_i = \min\{d_{ij}\}$, $\forall i, j \in E$.

Then, the spatial correlation coefficient between node i and node j is:

$$w_{ij} = \begin{cases} \dfrac{dm_i}{l_i d_{ij}}, & \{i, j\} \in E \\ 1 - \sum\limits_{\{i,j\} \in E} w_{ij}, & i = j \\ 0, & \text{otherwise} \end{cases} \tag{23}$$

According to the nature of the spatial correlation, the closer the two nodes are, the greater the influence of the data is on each other at the next moment, and the data spatial correlation would be greater. Based on the distance between nodes, Equation (23) can reasonably distribute the spatial correlation coefficient for adjacent nodes.

The correlation coefficient matrix is:

$$W = [w_{ij}] \in R^{n\times n} \tag{24}$$

where W is the one step transition probability matrix of Markov chain that is used in this paper to predict the spatial correlation data.

The stochastic process $\{S(t),\, t \in T\}$ and state space of $\{S_n : k > 0\}$ is I. For k values at time t, $t_1 < t_2 < \cdots < t_k$, $k \geq 3$, under the condition $S(t_k) = s_k$, $s_k \in I$, $k = 1, 2, \ldots, n-1$, the conditional distribution function of $S(t_k)$ is equal to that of $S(t_k)$ when $S(t_{k-1}) = S_{k-1}$, then, $\{S(t), t \in T\}$ is called the Markov process. The Markov process whose time and state in the process are discrete is a Markov chain [27,28]. In this paper, the monitoring data of sensor nodes' change process is a Markov chain. The chain's state space is denoted as $I = \{X(1), X(2), X(i), \ldots\}$, $X_i \in R^n$ where $X(t) = \{X_1(t), X_2(t), \ldots, X_n(t)\}$ which represents the data generated by n nodes at time t. For any positive integer k, there is:

$$P(k) = P(m, m+k) = P\{X(m+k)|X(m)\} \tag{25}$$

It is the k step transition probability matrix, by which Markov chain transfers from state $X(m)$ to state $X(m+k)$ through k step. If the transition probability $p_{ij}(m, m+k)$ is only related to nodes i, j and time interval, it could be denoted as $p_{ij}(k)$. This transition probability is with stationarity. When the Markov chain is homogeneous, the transition probability is:

$$p_{ij}(k) = p\{X_i(m+k)|X_j(m)\} \tag{26}$$

The k step transition probability matrix is:

$$P(k) = [p_{ij}(k)] \in R^{n\times n} \tag{27}$$

Here there are two properties: $0 \leq p_{ij}(k) \leq 1$, $\forall i, j \in T$ and $\sum_{j\in T} p_{ij}(k) = 1$, $\forall i \in T$, $n > 0$. One step transition probability of Markov chain is

$$p_{ij} = p_{ij}(1) = p\left\{X_i(m+1)|X_j(m)\right\} \tag{28}$$

One step transition probability matrix by one step transition probability is:

$$P = P(1) = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1j} & \cdots \\ p_{21} & p_{22} & \cdots & p_{2j} & \cdots \\ \vdots & \vdots & & \vdots & \\ p_{i1} & p_{i2} & \cdots & p_{ij} & \cdots \\ \vdots & \vdots & & \vdots & \end{bmatrix} \tag{29}$$

As we can see, the spatial correlation coefficient matrix W satisfies the two properties of the transition probability matrix $P(k)$. The correlation coefficient matrix W of the spatial correlation algorithm for data prediction satisfies the properties of the one step transition probability matrix of Markov chain. In this study, we use one step prediction method to predict data. The proposed data prediction algorithm is based on the transition probability matrix of Markov chain. According to the spatial correlation between nodes, we take the spatial correlation coefficient matrix W as one step transition probability matrix P, that is, $P = W$.

According to the current node change state of current data and the step transition probability matrix, the prediction formula is $\Delta X_i(t) = \sum_{j=1}^{n} p_{ij}\Delta X_j(t)$, $\forall i, j \in V$ where

t represents time, and

$$\Delta X_i(t) = X_i(t) - X_i(t-1) \tag{30}$$

$$X_i(t) = \begin{cases} X_i^p(t), & e_i(t)(\%) \le \varepsilon \\ X_i^m(t), & e_i(t)(\%) > \varepsilon \end{cases} \tag{31}$$

where $X_i^p(t)$ represents the predictive value of node i at time t, $X_i^m(t)$ represents the monitoring value of node i at t, ε represents the given error threshold, and $e_i(t)(\%)$ represents the prediction error percentage of the node i at time t. The calculation formula of the prediction error percentage of node i at time t is:

$$e_i(t)(\%) = \left| \frac{X_i^p(t) - X_i(t)}{X_i(t)} \right| \times 100 \tag{32}$$

Then, the predictive value of node i at time $t+1$ is:

$$\begin{aligned} X_i^p(t+1) &= X_i(t) + \Delta X_i(t+1) \\ &= X_i(t) + \sum_{j=1}^{n} p_{ij}(X_j(t) - X_j(t-1)), \quad \forall i, j \in V \end{aligned} \tag{33}$$

3.2. **Algorithm description.** The SASTC mainly includes two parts: the self-adaptive temporal correlation (SATC) prediction model based on grey prediction and the spatial correlation (SC) prediction model based on Markov chain. The working positions of the two prediction models are shown in Figure 7.

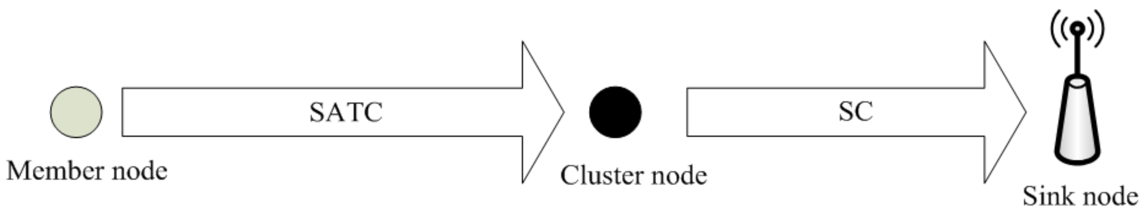

FIGURE 7. The working locations of the two models

The spatial correlation prediction model is used between cluster heads and the sink node, which uses the spatial correlation of the data among the cluster members to reduce the data transmission between cluster heads and the sink node. The algorithm is divided into the following three stages:

(1) In the network clustering stage, the system finds the cluster's Delaunay triangle adjacent graph.

(2) In the data collection stage of cluster heads, member nodes and cluster heads run the self-adapting time correlation prediction model to reduce the data transmission between member nodes and cluster heads.

(3) In the data collection stage of the sink node, member nodes and cluster heads run the spatial correlation prediction model based on Markov chain to reduce data transmission between cluster heads and the sink node.

The execution process of self-adapting spatial-temporal correlation data prediction algorithm is shown in Figure 8.

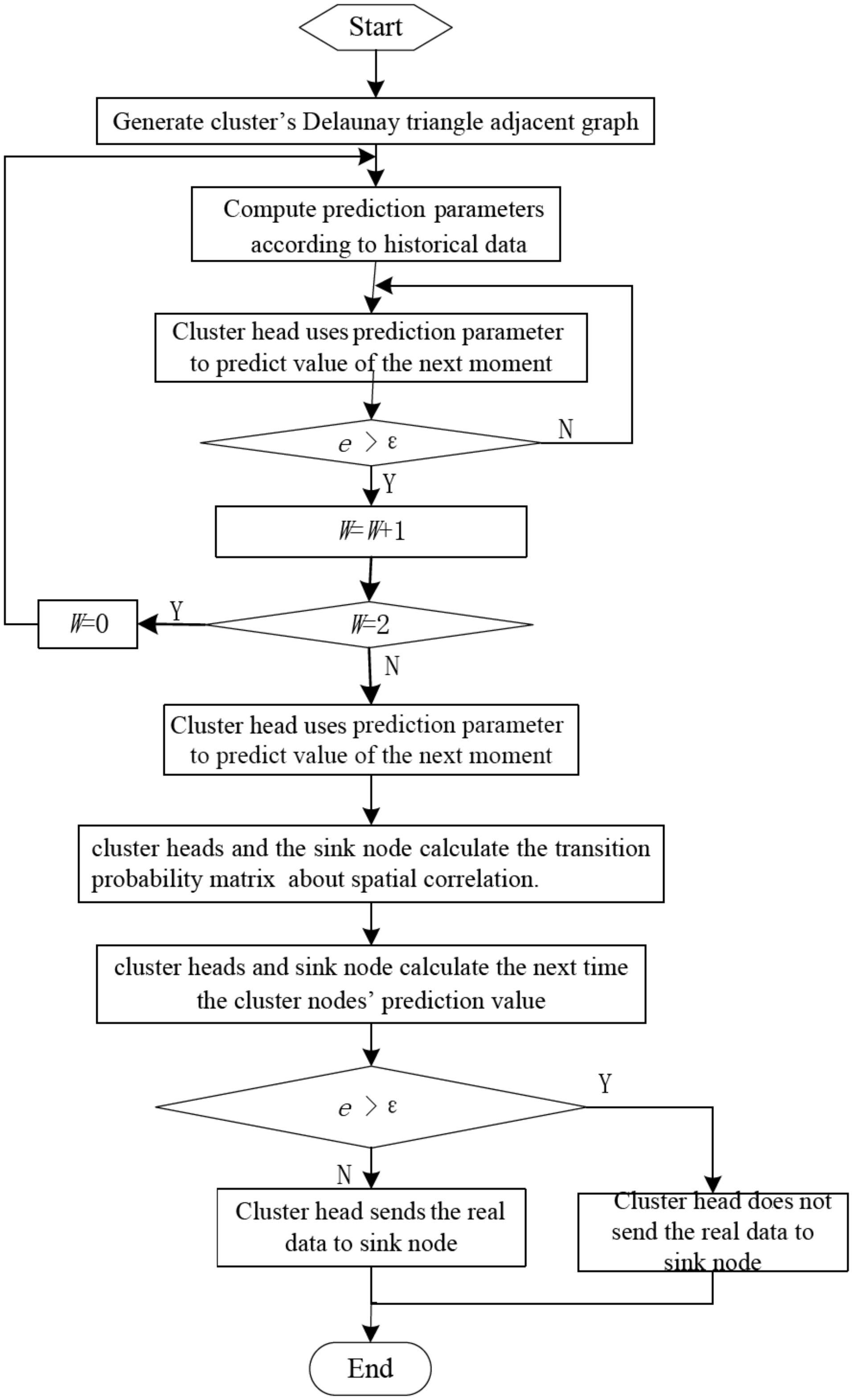

Figure 8. Flow chart of the integrated algorithm

4. Simulation and Analysis.

4.1. **Wireless communication energy consumption model.** In order to verify the performance of our proposed algorithm, this study uses Matlab simulation. In the 100m ∗ 100m region, we randomly deployed 100 nodes for energy consumption simulation. The simulation uses the LEACH protocol to cluster at first. The selection ratio of cluster head is 0.05. The model uses the method of single jump to transmit data. The Delaunay triangle adjacent graph is shown in Figure 1, which assume that the route between nodes with cluster heads and between cluster heads with the sink node are credible. Each node and cluster head transit data one time in every around. The time of nodes, cluster heads and the sink node is synchronous. The sink node and cluster heads use the unified data and the prediction algorithm. The prediction error threshold is 3% which is small enough for the actual applications.

In order to verify the data transmission algorithm and the network life cycle in the network, this paper adopts the data transmission model and the node energy consumption model to describe the data transmission and the node energy consumption situation in the network. The model parameters are shown in Table 1. These parameter values are recommended by TinyOS. TinyOS is especially useful for microcontroller-based devices that have sensors and/or networking capabilities. It is designed for resource-constrained devices such as microcontrollers and devices that need very low power. This research uses TinyOS as the research platform. The topology used in our simulation is the same as the real topology of the sensor network that will be deployed in one real garden in further study.

Table 1. Simulation parameters

Parameters	**Values**	**Unit**
Number of nodes	100	
Deployment scope	$(0,0)$ to $(100,100)$	m
Coordinates of the sink node	$(50,50)$	m
E_o	0.05	Joules
Packet length	4000	bit
E_{elec}	50	nJ/bit
ε_{mp}	0.0013	pJ/bit/m^4
ε_{fs}	100	pJ/bit/m^2
E_p	5	nJ/bit
Percentage of cluster head	0.05	
Prediction error threshold	0.03	

In Table 1, E_o represents the initial energy of the node, E_p represents the data process and the predicted energy consumption, E_{elec} represents the circuit energy consumption of sending or receiving data, and ε_{mp} and ε_{fs} represent the energy consumption of the signal amplifier, respectively. The nodes' distance is d, and the energy consumption for nodes sending n bits data is expressed as follows [23-25]:

$$E_{TX}(n,d) = \begin{cases} n \times E_{elec} + n \times \varepsilon_{mp} \times d^4, & d > d_0 \\ n \times E_{elec} + n \times \varepsilon_{fs} \times d^2, & d \le d_0 \end{cases} \tag{34}$$

where

$$d_0 = \left(\frac{\varepsilon_{fs}}{\varepsilon_{mp}}\right)^{\frac{1}{2}} \tag{35}$$

The energy consumption for nodes receiving n bits data is as follows:

$$E_{RX}(n) = n \times E_{elec} \tag{36}$$

The energy consumption for nodes processing and predicting n bits data is as follows:

$$E_P(n) = n \times E_P \tag{37}$$

The nodes' remaining energy is as follows:

$$E_r = E_o - (E_{TX}(n_1, d) + E_{RX}(n_2) + E_P(n_3)) \tag{38}$$

where E_r represents the nodes' residual energy, E_p represents the nodes' initial energy, n_1 is the total amount of data transmission, n_2 is the total amount of data reception, and n_3 is the total amount of data processing and prediction. The node dies when the node's residual energy $E_r \leq 0$.

4.2. **Simulation results and analysis.** Figure 9 shows the comparison of the SASTC algorithm, the ADC algorithm and no prediction algorithm. These algorithms are conducted with the error threshold being 3%. We can see the following observations.

(1) Among the three algorithms, the SASTC algorithm produces the least amount of data packets to the sink node. The SASTC algorithm can reduce about 3.74% of data transfer rate compared with the ADC algorithm. The transmission with no prediction algorithm is with the largest amount of data packets. This verified the effectiveness of SASTC and ADC for reducing data transmission in WSNs, and more importantly showed the advantage of our SASTC algorithm over the ADC algorithm.

(2) For all the three transmission algorithms, the simulation round to the stability is comparative, that is, about 800 rounds. This observation shows our SASTC algorithm does not take the efficiency as the cost to obtain the more performance in reducing the data transmission.

Figure 10 and Figure 11 show the compared results of dead node number and live node number during the simulation time, respectively. From the comparison in Figure 10, we can have the following observations.

(1) All the three algorithms have no dead nodes at the beginning, and have sudden increases in the number of dead nodes when the transmission rounds are over given values. This phenomenon is inevitable due to the energy consumption of sensing nodes.

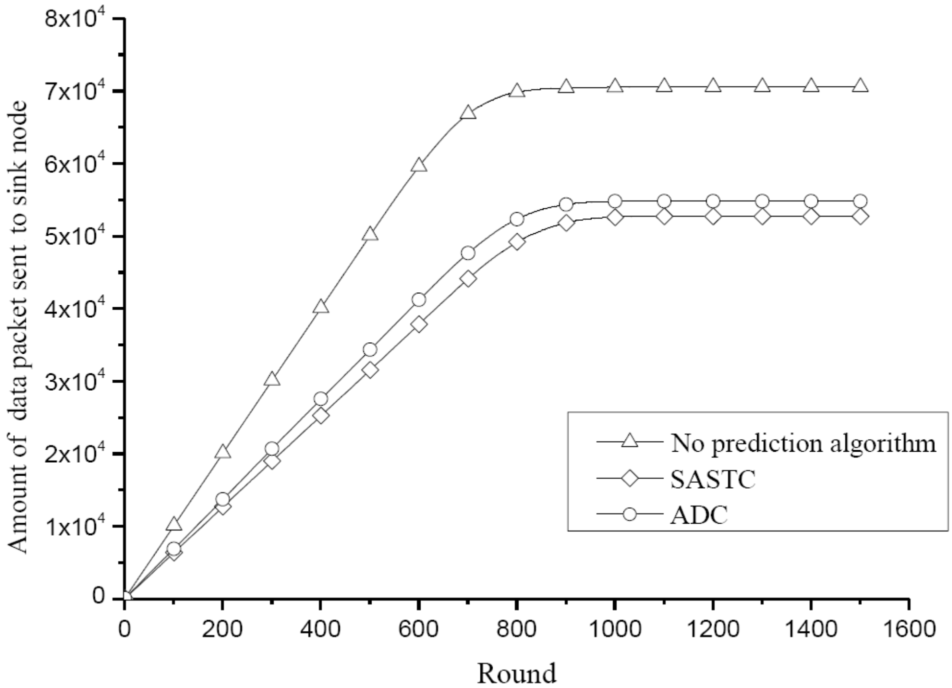

Figure 9. Comparison of data upload number

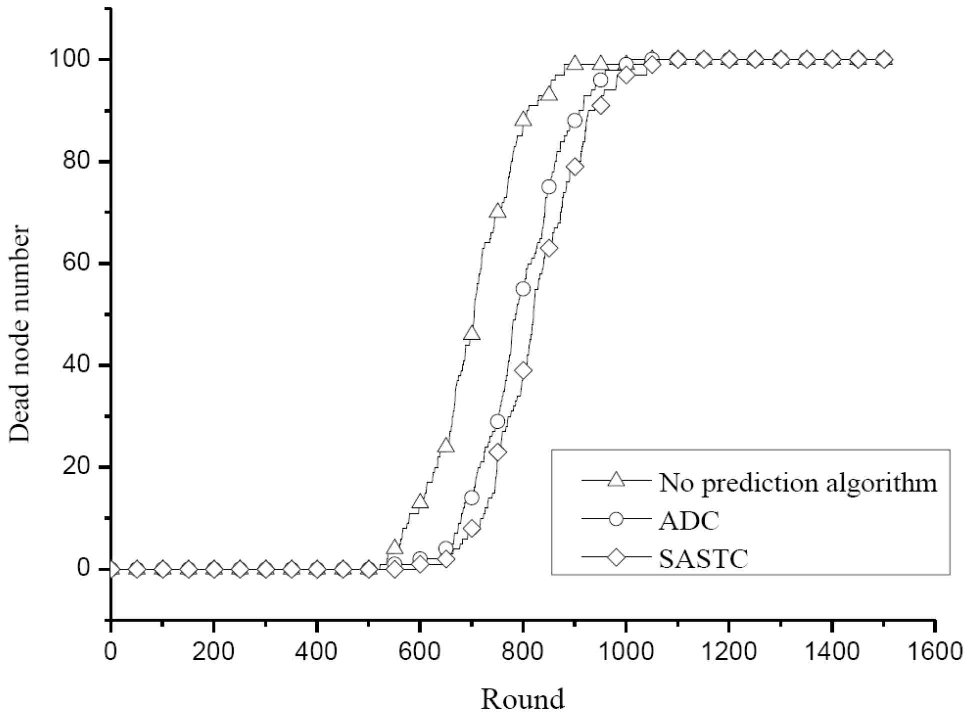

FIGURE 10. Comparison of dead node number

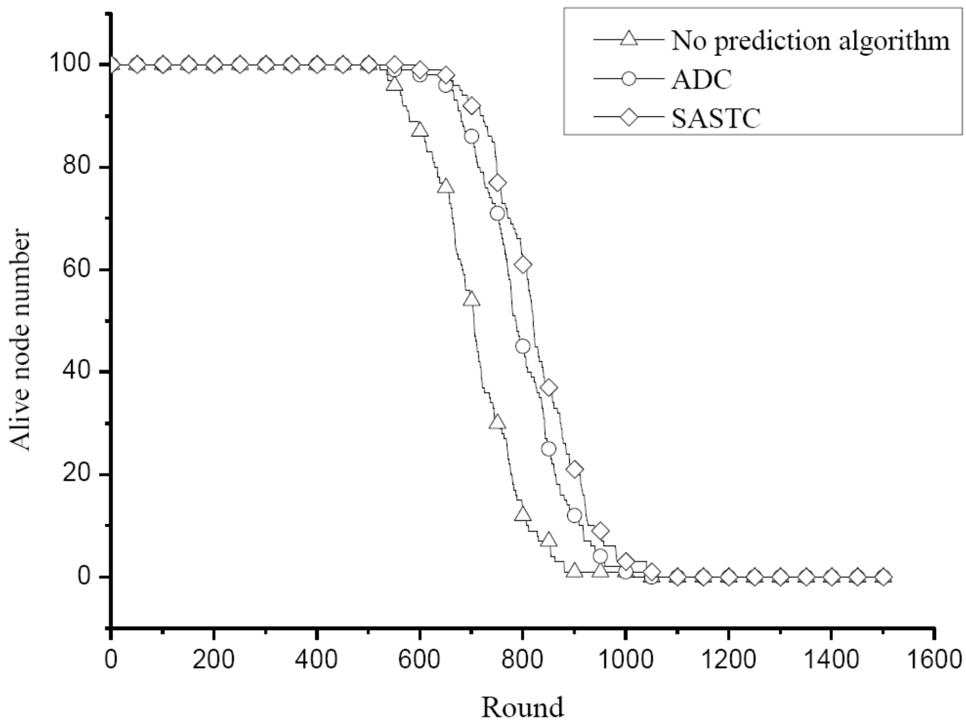

FIGURE 11. Comparison of live node number

(2) Although the three algorithms have the similar changing trends in the number of dead nodes, our SASTC algorithm shows the competitive advantage over other two algorithms. That is, our SASTC algorithm needs more rounds to reach to the same number of dead nodes.

From the comparison in Figure 11, we can have the following observations.

(1) All the three algorithms have 100% of live nodes at the beginning, but have sudden decreases in the number of live nodes when the transmission rounds are over given values. Similarly, this phenomenon is inevitable due to the energy consumption of sensing nodes.

(2) Although the three algorithms have the similar changing trends in the number of live nodes, our SASTC algorithm shows the competitive advantage over other two algorithms. That is, our SASTC algorithm has the most live nodes among the three algorithms.

Through the above simulation results, we can see that using SASTC algorithm is better than using ADC algorithm. The former can decrease network data transmission, reduce the energy consumption of the nodes data transmission and prolong the life cycle of the network.

5. **Conclusions.** In this paper, we applied the spatial-temporal correlation of WSNs data to the data prediction, and built the self-adapting linear prediction model and the spatial correlation prediction model based on the Markov chain. Through analyzing the two models, we proposed the SASTC prediction algorithm. This algorithm not only overcomes the low precision problem of the time correlation data prediction algorithm, but also reduces the requirement of data prediction algorithm for node's computing power and data storage capacity. The simulation results show that, compared with the ADC algorithm, the SASTC algorithm has higher prediction accuracy, can effectively reduce the amount of data transmission in the network and save the energy consumption of data transmission. For the future research, we plan to implement and evaluate our work in real sensor networks. The algorithm could be used in some aspects that have not been addressed in this work, such as leveraging the SASTC prediction algorithm to improve the coverage of wireless sensor networks.

REFERENCES

[1] Y. Bai, S. Liu, Z. Zhang and J. Yuan, EBTM: An energy-balanced topology method for wireless sensor networks, *International Journal of Innovative Computing, Information and Control*, vol.13, no.5, pp.1453-1465, 2017.

[2] I. Lazaridis and S. Mehrotra, Capturing sensor-generated time series with quality guarantees, *Proc. of the 19th Int. Conf. Data Eng.*, pp.429-440, 2003.

[3] M. Mollanoori, M. M. Hormati and N. M. Charkari, An online prediction framework for sensor networks, *Proc. of the 16th Iranian Conf. Elect. Eng.*, 2008.

[4] J.-J. Lim and K. G. Shin, Energy-efficient self-adapting online linear forecasting for wireless sensor network applications, *Proc. of the 2nd IEEE Int. Conf. Mobile Ad Hoc Sensor Syst.*, pp.372-379, 2005.

[5] E. I. Gaura, J. Brusey, M. Allen, R. Wilkins, D. Goldsmith and R. Rednic, Edge mining the Internet of things, *IEEE Sensors J.*, vol.13, no.10, pp.3816-3825, 2013.

[6] Q. Han, S. Mehrotra and N. Venkatasubramanian, Energy efficient data collection in distributed sensor environments, *Proc. of the 24th IEEE Int. Conf. Distrib. Comput. Syst.*, pp.590-597, 2004.

[7] S. M. McConnell and D. B. Skillicorn, A distributed approach for prediction in sensor networks, *Proc. of the 1st Int. Workshop Data Mining Sensor Netw.*, pp.28-37, 2005.

[8] D. P. Singh, V. Bhateja and S. K. Soni, Energy optimization in WSNs employing rolling grey model, *International Conference on Signal Processing and Integrated Networks (SPIN)*, pp.801-808, 2014.

[9] X. Wang, J.-J. Ma, S. Wang et al., Time series forecasting energy-efficient organization of wireless sensor networks, *Sensors*, vol.7, no.9, pp.1766-1792, 2007.

[10] X. Luo, J. Liu, D. Zhang et al., An entropy-based kernel learning scheme toward efficient data prediction in cloud-assisted network environments, *Entropy*, vol.18, no.8, p.274, 2016.

[11] L. Zhao, N. Chen and Y. Jia, An improved energy efficient routing protocol for heterogeneous wireless sensor networks, *International Journal of Innovative Computing, Information and Control*, vol.13, no.5, pp.1637-1648, 2017.

[12] Y. Wu, W. Liu and K. Li, A novel wireless acoustic emission sensor system for distributed wooden structural health monitoring, *International Journal of Innovative Computing, Information and Control*, vol.13, no.4, pp.1289-1306, 2017.

[13] L. A. Villas, A. Boukerche and D. Guidoni, A spatial correlation aware algorithm to perform efficient data collection in wireless sensor networks, *Ad Hoc Networks*, vol.12, no.1, pp.69-85, 2014.

[14] H.-B. Jiang, S.-D. Jin and C.-G. Wang, Prediction or not? An energy-efficient framework for clustering-based data collection in wireless sensor networks, *IEEE Trans. Parallel & Distributed Systems*, vol.22, no.6, pp.1064-1071, 2011.

[15] S. Goel and T. Imielinski, Prediction-based monitoring in sensor networks: Taking lessons from MPEG, *ACM SIGCOMM Comput. Commun. Rev.*, vol.31, no.5, pp.82-98, 2001.

[16] S. Goel, A. Passarella and T. Imielinski, Using buddies to live longer in a boring world, *Proc. of the 4th Annu. IEEE Int. Workshop Sensor Netw. Syst. Pervasive Comput.*, pp.342-346, 2006.

[17] K. Choi, S. H. Lim and J. H. Kim, Cost-effective monitoring algorithm for cyber-physical system platform using combined spatio-temporal model, *Journal of Supercomputing*, pp.1-12, 2016.

[18] S. Chen, C. Zhao, M. Wu et al., Compressive network coding for wireless sensor networks: Spatio-temporal coding and optimization design, *Computer Networks*, vol.108, pp.345-356, 2016.

[19] S. Kandukuri, J. Lebreton, N. Murad et al., Data window aggregation techniques for energy saving in wireless sensor networks, *Computers and Communication*, pp.226-231, 2016.

[20] G. Li, B. He, H. Huang et al., Temporal data-driven sleep scheduling and spatial data-driven anomaly detection for clustered wireless sensor networks, *Sensors*, vol.16, no.10, 2016.

[21] H. Edelsbrunner, *A Short Course in Computational Geometry and Topology*, Springer, 2014.

[22] J. Boissonnat, R. Dyer and A. Ghosh, The stability of Delaunay triangulations, *International Journal of Computational Geometry & Applications*, vol.23, pp.303-333, 2014.

[23] N. Rubin, On topological changes in the Delaunay triangulation of moving points, *Discrete & Computational Geometry*, vol.49, no.4, pp.710-746, 2013.

[24] T. Komorowski, S. Peszat and K. Szare, On ergodicity of some Markov processes, *Annals of Probability*, vol.38, no.4, pp.1401-1443, 2008.

[25] C. Wang, H. Ma, Y. He et al., Adaptive approximate data collection for wireless sensor networks, *IEEE Trans. Parallel & Distributed Systems*, vol.23, no.6, pp.1004-1016, 2012.

[26] G. Anastasi, M. Conti, D. Francesco et al., Energy conservation in wireless sensor networks: A survey, *Ad Hoc Networks*, vol.7, no.3, pp.537-568, 2009.

[27] M. Arns, P. Buchholz and A. Panchenko, On the numerical analysis of inhomogeneous continuous-time Markov chains, *Informs Journal on Computing*, vol.93, no.4, pp.612-615, 2010.

[28] A. Y. Zomaya and W.-S. Si, New memoryless online routing algorithms for Delaunay triangulations, *IEEE Trans. Parallel & Distributed Systems*, vol.23, no.8, pp.1520-1527, 2012.